XStatic: A Novel ESD Verification and Debug Platform

National Semiconductor

Rajesh R. Berigei
Ganesh R. Shamnur

Session 6.8

Presented at

cadence designer network

cdnLIVE

Silicon Valley 2007
Abstract

Electrostatic discharge is defined as the transfer of charge between bodies at different electrical potentials. However, electrostatic discharge can change the electrical characteristics of a semiconductor device, degrading or destroying it. ESD being a high surge current event has become a major reliability problem in the semiconductor industry. ESD Testers are commercially available in the market, where IC’s are tested for their ESD performance. These conventional approaches are post-fabrication methods which leave very narrow design time window for rectifying the ESD problems. So, capturing this predicament at an early phase of IC design would provide the designer more flexibility to undertake necessary precautions and help achieve ESD proof designs. This was the motivation to come up with an in-house ESD verification platform that aids in debugging complex ESD situations by checking for appropriate guidelines and by performing ESD simulations. Due to the complexity associated with ESD protection design, a structured approach using a suite of CAD tools would be of great value. This work describes a tool framework that will help circuit designers identify potential limitations of the ESD protection scheme and fix them prior to finalizing the floorplan for any given design, as well as serving as a design debug environment to help identify ESD related limitations for circuits failing ESD tests. This paper will describe the ESD verification platform at National Semiconductor and explains how it has helped analog designers at National to come to grips with ESD related design issues.

Keywords: ESD, ElectroStatic Discharge, HBM, MM, CDM, TLP.

1. Introduction

ESD protection mechanisms are built with the main objective to shunt ESD currents through predefined harmless discharge paths. To locate and fix the appropriate current paths that might fail for ESD is the other important task of the designer.

1.1 Common ESD problems

ESD is becoming an increasingly complex problem with the introduction of deep submicron technologies. With technology scaling, supply voltages are lowered, channel lengths, junction depths and gate-oxide thickness reduced. With these trends of miniaturization proceeding, the risk of IC malfunctioning on account of ESD is on the rise. Some of the various reasons for failure are: wrong selection of protection circuits, non-optimum metal connections, improper protection circuit connection, incorrect bus layout and non-intentional layout errors. With the drive to achieve higher chip densities,
many fatal mistakes are committed by the designers which would be realized only after the chip fabrication. ESD protection design is an experimental process wherein test matrices with several variations of protection devices are tried and characterized to finalize on the optimum circuit. Thereby, these iterations would leave a very narrow ‘market window’.

1.2 CAD tool approach

Therefore, an ideal approach is to make checks for ESD guidelines compliance and warn designers about the ESD design errors through a software utility before finalizing on the chip floor-plan. A guideline based ESD tracking method and a simulation approach with analysis facility is the idea behind the proposed CAD platform that allows designers to address ESD failures caused not only due to layout errors, but also by rather due to inappropriate protection design in the I/O cell ring.

1.3 XStatic Components

The XStatic, ESD debug environment comprises of two main elements.

a) XStatic Static Checker: This looks at a chip layout to identify violations specific to ESD design guidelines. This is also known as ESDIO Guideline Checker utility (ESDIOGC) where checks for ESD guideline compliance to chips are made. The guideline checker reads the technology information for the design along with the design database and performs a spacing based ESD rule checking.

b) XStatic Dynamic Checker: This is a dynamic ESD verification and debug environment that utilizes circuit simulation to identify ESD related issues. Given the schematic/layout of an IO-ring, various sets of zap pairs (based on the terminals in the schematic/layout) are created and simulated with different ESD stimuli. In case of schematic designs, an intermediate ‘schematic with embedded-resistances’ view of the design is created to make the analysis more realistic. A parasitic extracted view of the design is used in case of layout simulations. Results are captured at key probe points during simulations, which are compared with the threshold values specified by the
designers to rank a particular zap pair simulation as either passed (voltage less than breakdown voltage) or violated (voltage greater than breakdown voltage) for the ESD event. A back-annotation mechanism would highlight the actual inner devices of failure on account of the ESD stimulus. Thereby, designers can analyze the failure to identify and fix the cause of ESD problem in the circuit.

2. ESD Tool Architecture and Functionality

Figure 1 shows a high level Xstatic flow where the XStatic Static and Dynamic checker modules are shown with the data-flow diagram.

Fig 1: XStatic Architecture

2.1 XStatic Static Guidelines Checker

The Xstatic static checker gets design data directly from a design database and also from a technology specific guideline database. This technology guideline file contains process breakdown data as well as key electrical characteristics of the ESD
devices to be used. The checker generates a guidelines violations file. The violations can be annotated into the layout editor by means of an ESD annotator utility.

It should be noted that the Xstatic Static Guidelines checker applies to an ESD scheme that has a regularly arranged IO pad ring. The Static checker flow is as shown in the Fig 2. For chips that have IO devices not arranged in the shape of a ring, the static guideline checking is skipped. The tool initially reads the layout database and the technology dependent guidelines file. It then traverses each IO cell in a specific direction. For each IO cell that is encountered the tool determines whether the cell is a protection cell or not. The tool computes the number of ESD protection cells that are in the vicinity of each IO cell and provides an assessment whether the ESD guidelines have been satisfied or violated.

2.2 XStatic Dynamic ESD Verification and Debug Environment

The Xstatic dynamic checker gets input from the schematic/layout database. It generates an ESD verification database, which is read by a database post processor, and which identifies the ESD breakdown voltage violations to generate information about the ESD high resistance paths. This data is then back annotated into the schematic/layout editor.

The ESD verification environment acts as a “virtual” ESD test setup. Given the schematic/layout of an IO-ring, XStatic creates various sets of zap pairs (based on the terminals in the schematic/layout) to be tested with a specified ESD stimulus. Xstatic-I works only on schematic designs. It provides for a “what-if” design scenario by working
on estimates of bus resistances. Xstatic-II works on real layouts. It invokes Assura-RCX and runs simulations using actual parasitics. Transient simulations with the User specified error preset, stop time and simulation options can be performed. XStatic thus tries to replicate the actual on-field ESD events on the design. Various probe types can be analyzed in a single run. The results are then compared with voltage threshold specified by the designer to rank a particular zap pair ESD event as passed (voltage less than pre-specified breakdown voltage) or violated (voltage greater than pre-specified breakdown voltage). Designers can thereby analyze these results to identify the root cause of the ESD failures and fix them. The flow is as shown in Fig 3.

![Fig 3: XStatic Dynamic Checker Flow](image)

XStatic also provides debugging interface to ESD simulations, by suitably displaying the simulation results and by back-annotating onto the schematic design. Designer has an option to choose the zapPairs that one intends to simulate (all ZapPairs or user-selected ZapPairs). The tool gives provision to set various probe-points in the design. It facilitates the designer to hook-up an HBM/MM source or a stimulus of his/her choice for simulations. Support to local (sequential) and distributed (parallel) modes of simulation has been provided. Provides an interactive histogram to enable back-annotation for passed and violated simulations. Provision to re-use previous netlists is provided. Internally generates a parasitic resistance embedded schematic design for XStatic-I simulations. Provision to include bondwire inductance effects during ESD simulations has been provided. A set of all possible pairs of pins on the chip
is computed. One pin of each pair is stressed with an ESD voltage and the other pin is grounded. These zap-pairs are auto-netlisted out and simulated. Prior to simulation, the layout is extracted to get accurate parasitics in case of XStatic-II which is accounted during simulations.

2.3 Current Path Illustration

Fig 4 shows a currents path illustration, where there are two protection cells such as Clamps cells and three non-protection IO cells, with one of them being a signal IO cell and the other two being VDD and VSS IO cells. The diodes are not shown, but these are assumed to be inside the IO cells. Since there are three pads and two protection cells, there could be twenty-four different paths for six different zap pairs. For example, if the zap pair under consideration is (IO, VSS), there are four different paths. If an ESD event were to occur at IO, the current will branch through these paths. Acw is the anticlockwise branch and Cw is clockwise branch.

Possible Paths are:

i) Acw(IO-DIODE-MC1-DIODE-VSS)
ii) Cw(IO-DIODE-MC1-DIODE-VSS)
iii) Acw(IO-DIODE-MC2-DIODE-VSS)
iv) Cw(IO-DIODE-MC2-DIODE-VSS)

Similarly there will be four paths each between IO-VDD, VSS-VDD, VSS-IO, VDD-VSS and VDD-IO. Each of these current paths will have a different current flowing through it.
3. XStatic Results/Analysis

Below are some result samples that XStatic would output to the designer for ESD debugging.

3.1 Static Checker Results Summary:

The static checker outputs would be based upon the layout rules specified in the design-rule file and the back-annotation utility is as shown in the Fig 5.

**Rule 1: Error** Incorrect domain connection of ESD protection cell.

   e.g. if VDDMC is connected to VDDIO.

**Rule 2: Information** Found more ESD protection cells than required.

**Rule 3: Warning** Potential imbalance in ESD protection cells. Found X ESD cells on one side and Y on the other.

**Rule 4: Error** Insufficient number of ESD protection cells found for a particular pin.
3.2 Dynamic Checker Results:

A job-monitor interface is provided with this to view the esd-passed, esd-violated and simulation-failed pin pairs. The esd-violated form is as shown in Fig 6.
The measured voltage along with the designer entered breakdown threshold voltage for the selected probe-variant is shown in the figure. The tool User can select a pin-pair and wish to probe its internal devices which would pop-open an interface as in Fig 7 which would assist in tracking the ESD failure devices. Selection of an internal device would highlight the device in Schematic/Layout editor the appropriate elements that formed the part of pin-pair current path.

![Fig 7: Internal Devices Probe Form](image)

4. Conclusions

The development of Xstatic is a cross functional effort between ESD and CAD to develop a software engine for assessing the viability of any ESD scheme as implemented in a specific circuit. The tool is meant to be useful both for regularly arrayed ESD cells and designs using a more random arraying. Some of the key accomplishments of this tool are:

i) Identify voltages that exceed a designer specified maximum voltage during a simulated ESD event

ii) Allows efficient management of simulating pin to pin ESD stress

iii) Using voltages at pads, XSTATIC overestimates protection required for a more robust design.
iv) XSTATIC I helps to correctly size metal bussing and via/contact count prior to initial layout. Since XSTATIC I uses line length in schematic to model interconnect resistance, estimates of maximum resistance can be calculated.

v) Identifies lack of protection devices.

vi) Identifies incorrect metal routing to pin bypassing ESD protection.

vii) Determine design limitations bondwires place on ESD protection.

viii) XSTATIC has the ability to run targeted reduced pin simulation to isolate failing pins.

ix) Reduces Silicon iterations due to ESD failures.

5. Acknowledgements

The authors would like to acknowledge Guruprasad Katti for his involvement in the project and also thanks to Sury Maturi, David Tamura and Andy Franklin for their guidance and vision in the project. Acknowledgement also goes to Don Archer and George Cleveland for their initial feedback on the requirements from a product line perspective.